

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

# PK0

Zusatzkurs für Programmieranfänger im WS 16/17

Marvin Gülzow

Universität Konstanz

2016-11-02

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

## 1 Kontext

- Transistoren
- Prozessor
- Programme
- JVM

## 2 Statements und Bedingungen

- Statements
- Evaluation
- Bedingungen

# Transistoren

PK0

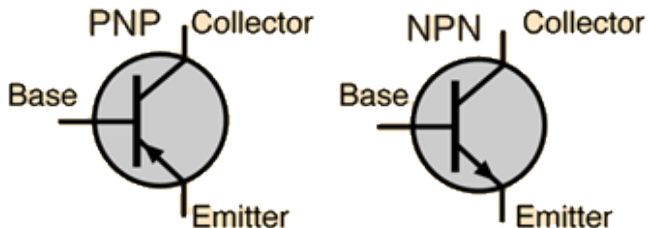
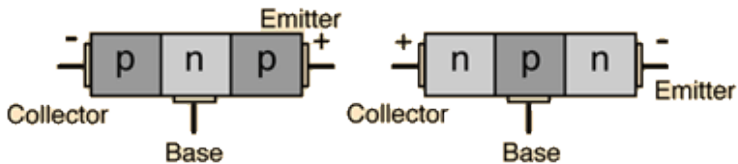
Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen



# Transistoren

PK0

Marvin  
Gülzow

Kontext

Transistoren

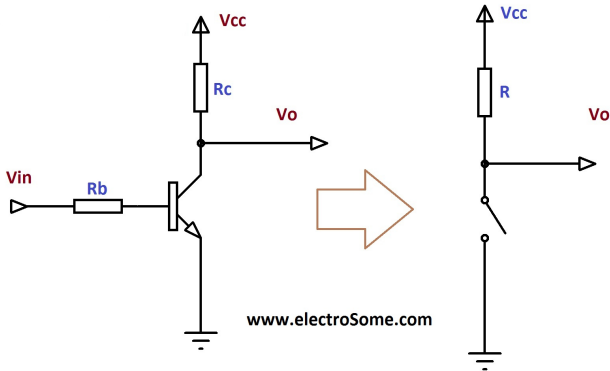
Prozessor

Programme

JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen



# Transistoren

PK0

Marvin  
Gülzow

Kontext

Transistoren

Prozessor

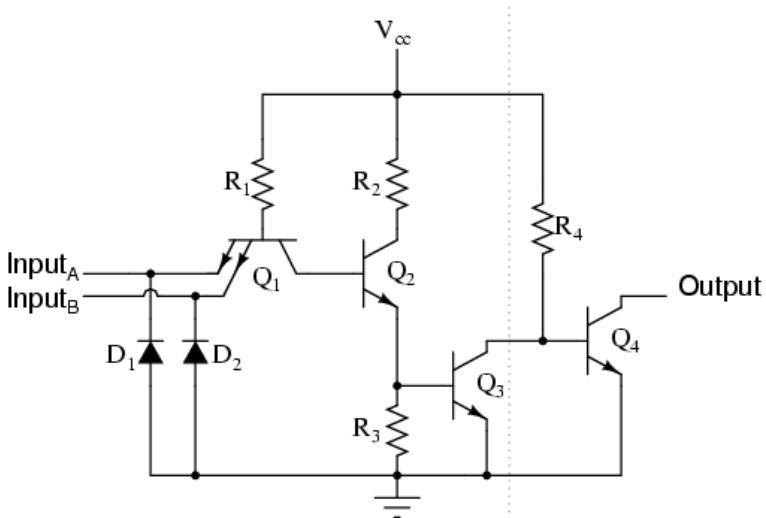
Programme

JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

*AND gate with open-collector output*



# Transistoren

PK0

Marvin  
Gülzow

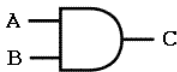
Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

AND



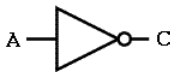
| Inputs |   | Output |
|--------|---|--------|
| A      | B | C      |
| 0      | 0 | 0      |
| 0      | 1 | 0      |
| 1      | 0 | 0      |
| 1      | 1 | 1      |

OR



| Inputs |   | Output |
|--------|---|--------|
| A      | B | C      |
| 0      | 0 | 0      |
| 0      | 1 | 1      |
| 1      | 0 | 1      |
| 1      | 1 | 1      |

NOT



| Input | Output |
|-------|--------|
| A     | C      |
| 0     | 1      |
| 1     | 0      |

# Transistoren

PK0

Marvin  
Gülzow

Kontext

Transistoren

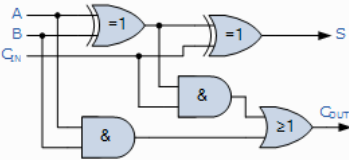
Prozessor

Programme

JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

| Symbol  | Truth Table |   |   |     |       |
|---|-------------|---|---|-----|-------|
|  | C-in        | B | A | Sum | C-out |
|   | 0           | 0 | 0 | 0   | 0     |
|   | 0           | 0 | 1 | 1   | 0     |
|   | 0           | 1 | 0 | 1   | 0     |
|   | 0           | 1 | 1 | 0   | 1     |
|   | 1           | 0 | 0 | 1   | 0     |
|   | 1           | 0 | 1 | 0   | 1     |
|   | 1           | 1 | 0 | 0   | 1     |
|   | 1           | 1 | 1 | 1   | 1     |

# Prozessor

PK0

Marvin  
Gülzow

Kontext

Transistoren

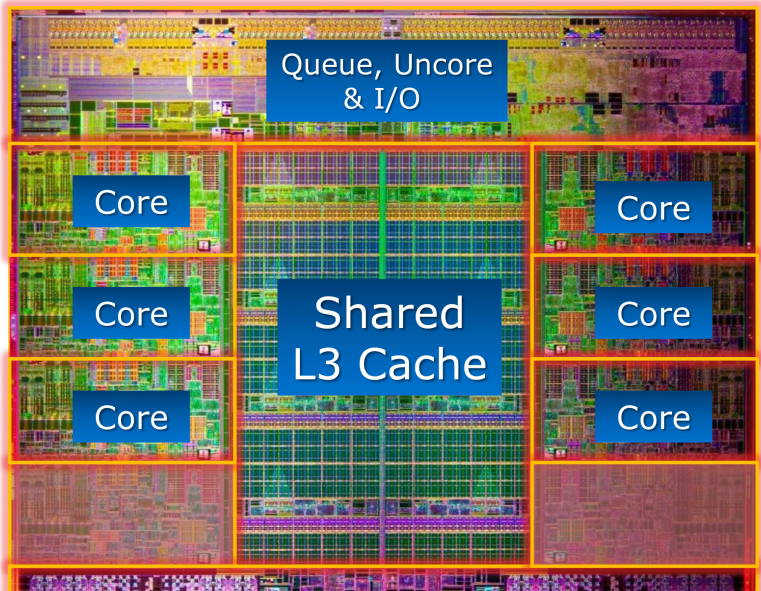
**Prozessor**

Programme

JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen





# Prozessor

PK0

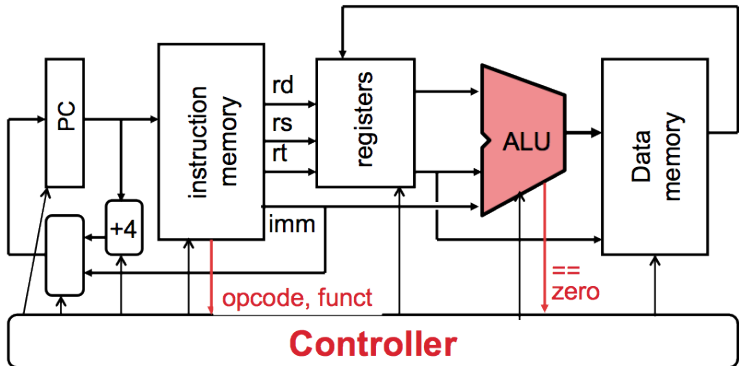
Marvin  
Gülzow

Kontext

Transistoren  
**Prozessor**  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen



# Prozessor

PK0

Marvin  
Gülzow

Kontext

Transistoren

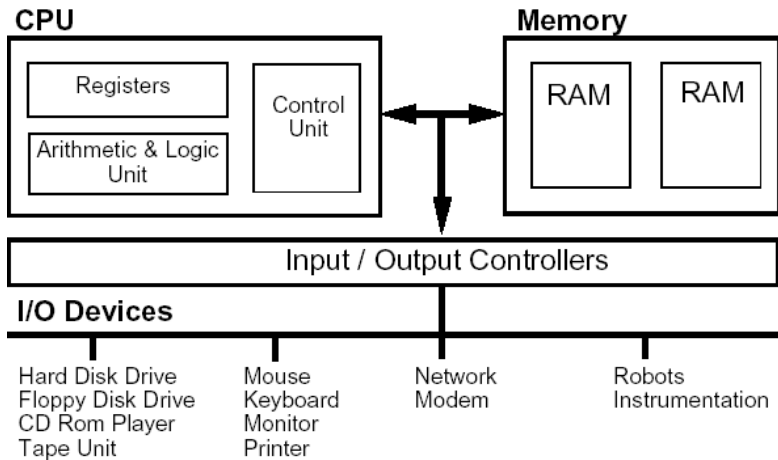
**Prozessor**

Programme

JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen



# Programme

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
**Programme**  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
_start:  
mov cx, 10  
mov esi, 0  
loop:  
inc esi  
dec cx  
jnz loop
```

⇒ b90a 0066 be00 0000 0066 4649 75fb

# Programme

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int c = 10;
```

```
int e = 0;
```

```
while(c > 0) {  
    ++e; // e = e + 1  
    --c;  
}
```

```
_start:  
mov cx, 10  
mov esi, 0  
loop:  
inc esi  
dec cx  
jnz loop
```

# Programme

PK0

Marvin  
Gülzow

Kontext

Transistoren

Prozessor

Programme

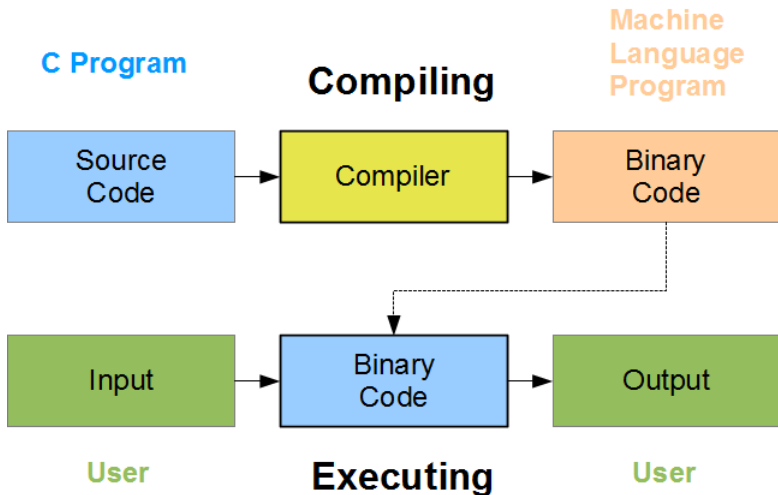
JVM

Statements  
und  
Bedingungen

Statements

Evaluation

Bedingungen



# JVM

PK0

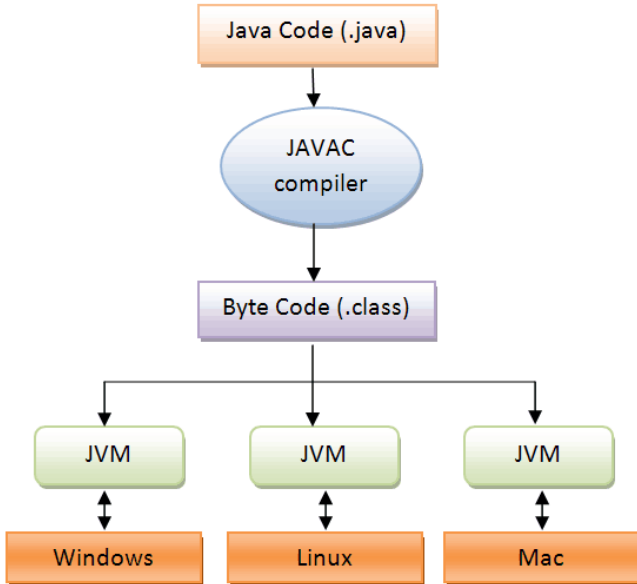
Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen



# JVM

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;  
i += 1;  
i = i * 2
```

```
iconst_0      // 03  
istore_0      // 3b  
iinc 0, 1     // 84 00 01  
iload_0       // 1a  
iconst_2      // 05  
imul          // 68  
istore_0      // 3b  
goto -7       // a7 ff f9
```

⇒ 03 3b 84 00 01 1a 05 68 3b a7 ff f9

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

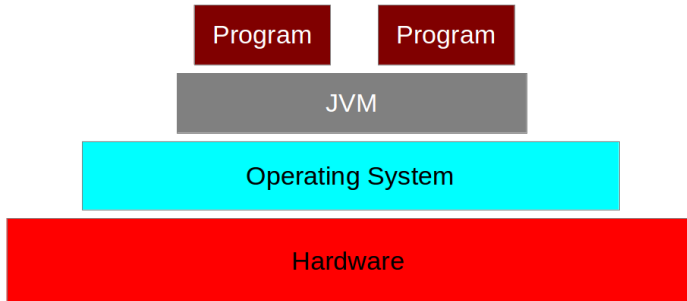
Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

- 1 Javacode schreiben
- 2 Bytecode
- 3 JVM
- 4 Bytes im RAM
- 5 Elektronen in der CPU
- 6 Quantenmechanik



## Java Virtual Machine (VM)



# Statements

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;
```

```
int x = 1;
```

```
i = 1;           // 1
```

```
i = 2;           // 2
```

```
i = i + 1;       // 3
```

```
i = i + x;       // 4
```

```
i += 1;          // 5
```

```
i += x;           // 6
```

```
++i;              // 7
```

```
i++;              // 8
```

# Statements

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;
```

```
int x = 1;
```

```
i = i + 2; // 2
```

```
i = i - 1; // 1
```

```
i = i * 2; // 2
```

```
i = i / 2; // 1
```

```
i = i + 2; // 3 :)
```

```
i = i / 2; // 1 WTF?
```

# Statements

PK0

Marvin  
Gülzow

## Kontext

Transistoren  
Prozessor  
Programme  
JVM

## Statements und Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;
```

```
int x = 1;
```

```
i = i + x;      i += x;    // Addition
```

```
i = i - x;      i -= x;    // Subtraktion
```

```
i = i * x;      i *= x;    // Multiplikation
```

```
i = i / x;      i /= x;    // Division
```

```
i = i % x;      i %= x;    // Modulo
```

# Statements

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;
```

```
++i;
```

```
i += 1;
```

```
i++;
```

```
int tmp = i;
```

```
i += 1;
```

```
return tmp;
```

# Statements

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;
```

```
int x = 0;
```

```
x = ++i;    // x == 1 == i
```

```
x = i++;    // x == 1; i == 2;
```

⇒ Benutzt ++i.

# Evaluation

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int a = 1 + 2 * 3;    // 7
int b = (1 + 2) * 3;  // 9
```

- Alles Klammern
- Klammern sind billig
- Keiner kann sich alle Operatoren merken

# Evaluation

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;  
int x = 1;
```

```
i == x;      // false  
i != x;      // true  
i < x;       // true  
i > x;       // false  
(i+1) <= x;  // true;  
i >= x;      // false;
```

```
boolean comparison = i == x;
```

```
comparison += 1; // Nein.
```



# Bedingungen

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;
```

```
int x = 1;
```

```
if (i == x) {  
    System.out.println("Equal.");  
}
```

```
if (i != x) {  
    System.out.println("Not equal.");  
}
```

# Bedingungen

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;
int x = 1;

if (i == x) {
    System.out.println("Equal.");
} else {
    // Fall (i != x)
    System.out.println("Not equal.");
}
```

# Bedingungen

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;
```

```
int x = 1;
```

```
if (i == x) {
```

```
    System.out.println("Equal.");
```

```
} else if (i != x) {
```

```
    System.out.println("Not equal.");
```

```
} else {
```

```
    System.out.println("CPU is broken.");
```

```
}
```

# Bedingungen

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;
```

```
int x = 1;
```

```
boolean comparison = (i == x);
```

```
if (comparison == true) {
```

```
    System.out.println("Equal.");
```

```
} else if (compariosn == false) {
```

```
    System.out.println("Not equal.");
```

```
} else {
```

```
    System.out.println("CPU is broken.");
```

```
}
```

# Bedingungen

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
int i = 0;
```

```
int x = 1;
```

```
boolean comparison = (i == x);
```

```
if (comparison) {
```

```
    System.out.println("Equal.");
```

```
} else if (!comparison) {
```

```
    System.out.println("Not equal.");
```

```
} else {
```

```
    System.out.println("CPU is broken.");
```

```
}
```

# Bedingungen

PK0

Marvin  
Gülzow

Kontext

Transistoren  
Prozessor  
Programme  
JVM

Statements  
und  
Bedingungen

Statements  
Evaluation  
Bedingungen

```
mov esa, 0
mov esb, 1
```

```
cmp i, x
je equal
jne notequal
```

```
print "CPU_ broken "
ret
equal:
print "Equal "
ret
notequal:
print "Notequal "
ret
```

- Alles ist kompliziert
- Alles basiert auf einem riesigen Technikstack
- Der Übergang von Ideen des Programmierers zu Elektronen in Silizium ist über Layer gelöst
- Die Technik ganz unten formt die Sprachen ganz oben
  - Ultraschnelles Addieren
  - Alles ist zahlenbasiert
  - Daten/Programm-Dualität
- Wer den ganzen Stack im Hinterkopfe hat versteht besser was er/sie/es tut
- Nächstes mal: Mehr praktisches :)